UW MMC Update

T. Gorski, D. Seemuth, A. Perugupalli,
A. Farmahini-Farahani

University of Wisconsin

May 23, 2012

UW MMC Version 2.0

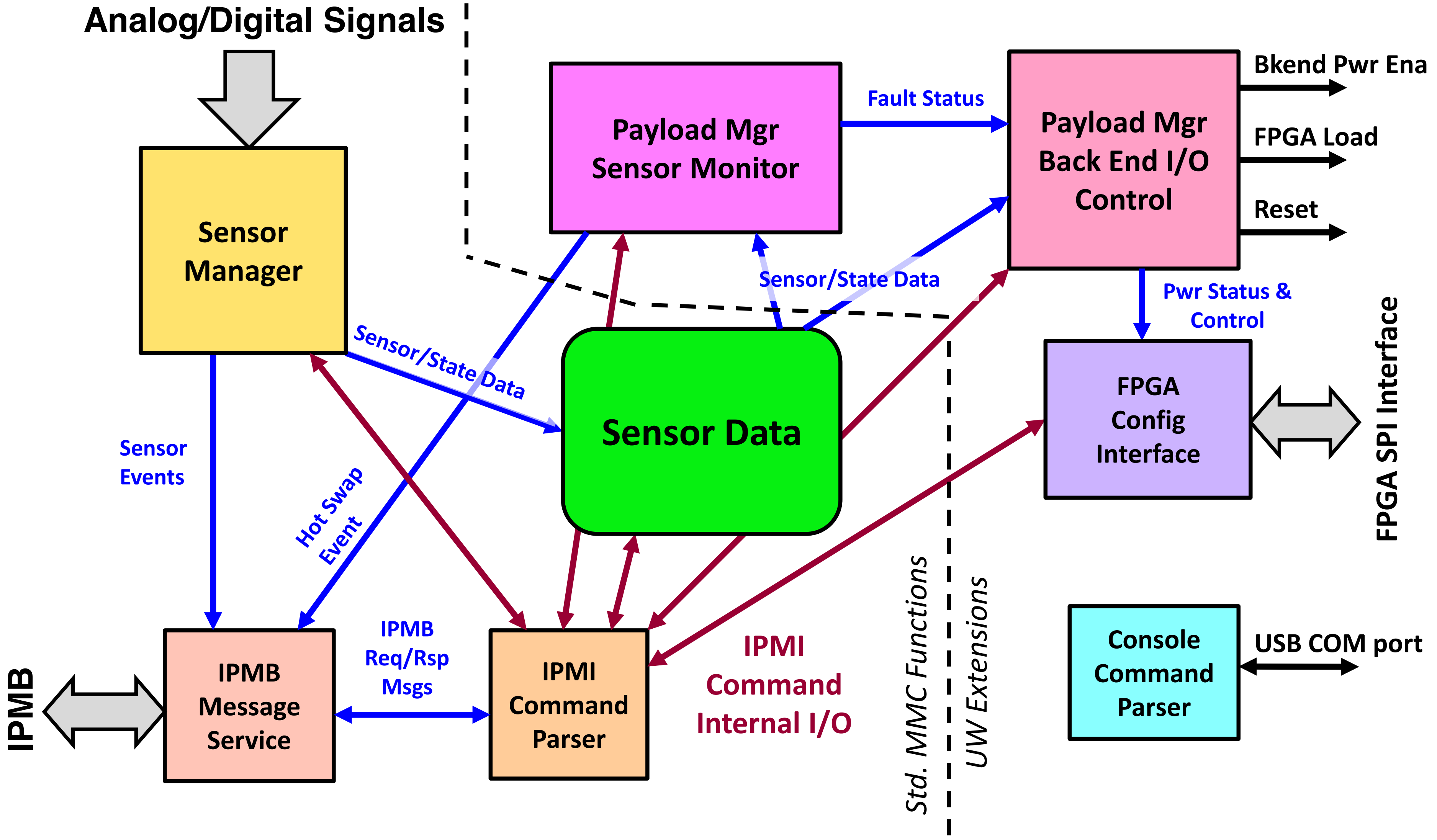
New Features:

- **8 additional analog sensors (for total of 16)**
  - CTP implementation: 14 voltages, 2 temperatures
- **3 new digital sensors**
  - Power Good, FPGA Load Done, FPGA Request
- **FPGA SPI Configuration Path**
  - Auto-detection, with local config data from MMC EEPROM or remote data from IPMI System Manager
  - Available Dual Port RAM based SPI HDL core
- **Multi-stage Back End Power Enable**
  - Up to 6 Separate Enables in up to 4 Stages, with programmable inter-stage delay
  - Settings stored in EEPROM

UW MMC Version 2.0, cont’d

- Initial Testing/Debug to be performed on UW Cal Trigger Processor (CTP) card
- Will port to BU AMC13
  - Reduced Sensor Model
  - Work with BU to integrate MMC-based Config Path into the standard firmware model
    - Anticipate including an IP address assignment mechanism
  - Release of new MMC via AMC13 web portal
- Expected availability: Summer, 2012

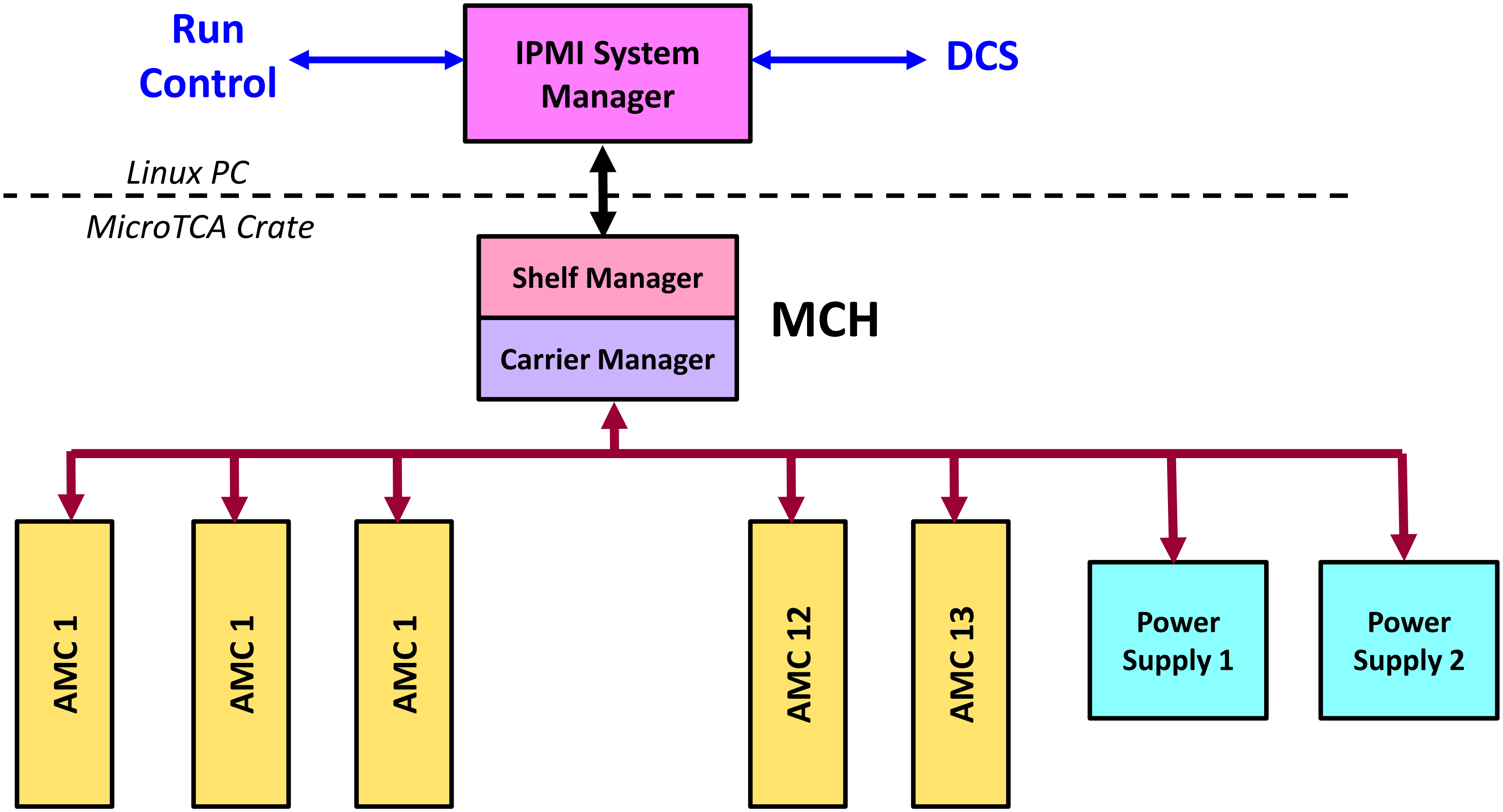
UW MMC “Simplified” Block Diagram



IPMI System Manager

- System Manager: A level of management functionality above the Shelf Manager(s), charged with the management of an entire subsystem
- Definition does not imply a specific implementation
  - Could mean independent IPMI System Managers for each crate or subsystem
  - Will depend on the LAN configurations employed
  - Implementation could be a common platform with subsystem-specific customizations, or not
- Provides service to both DCS and Run Control:
  - DCS: Environmental sensors (e.g. temperature)
  - Run Control: Subsystem-specific configuration & initialization/control path

Example IPMI Small System Block Diagram



UW IPMI System Manager R&D

- **Goals:**
  - **Fully support UW MMC IPMI Command Set**
  - **Connections as necessary to DCS and Trigger Supervisor**
  - **Ability to deliver sensor event notification to local subsystem-specific controls (e.g., FPGA Load Done or Power Good sensors)**
  - **Easily deployed by any/all users of UW MMC**
- **Form:**
  - **May be some combination of server-assisted and direct I/O (e.g., server for reservation and event notification, direct IPMI I/O between AMC card and subsystem control software for most commands)**
  - **Minimal use of database and/or config files (automatically configure from connected hardware as much as possible)**

Common IPMI Functionality

- Suggested common functionality (supported by all CMS MMCs)
  - Back end power remote on/off control
  - Cold reset and FPGA Firmware reload (via *PICMG FRU Control* command)
  - 2 Temperature Sensors (ambient, hot spot) with full threshold event support
  - Configuration I/O interface between MMC and FPGA(s)
    - Data sourced either from on-board nonvolatile storage (e.g. EEPROM) or via remote IPMI command
  - Remotely editable, non-volatile storage of sensor SDR (sensor definitions)

IPMI Command Network Function (netFN) Field

- 6-bit field identifying the set of functions to be accessed
- Requests have even value, responses have odd value (+1 greater)
- netFN values in the range 00h-2Fh are already defined or reserved by IPMI
- netFN values 30h-3Fh are available for OEM applications
- UW MMC custom commands use netFN code 32h-33h

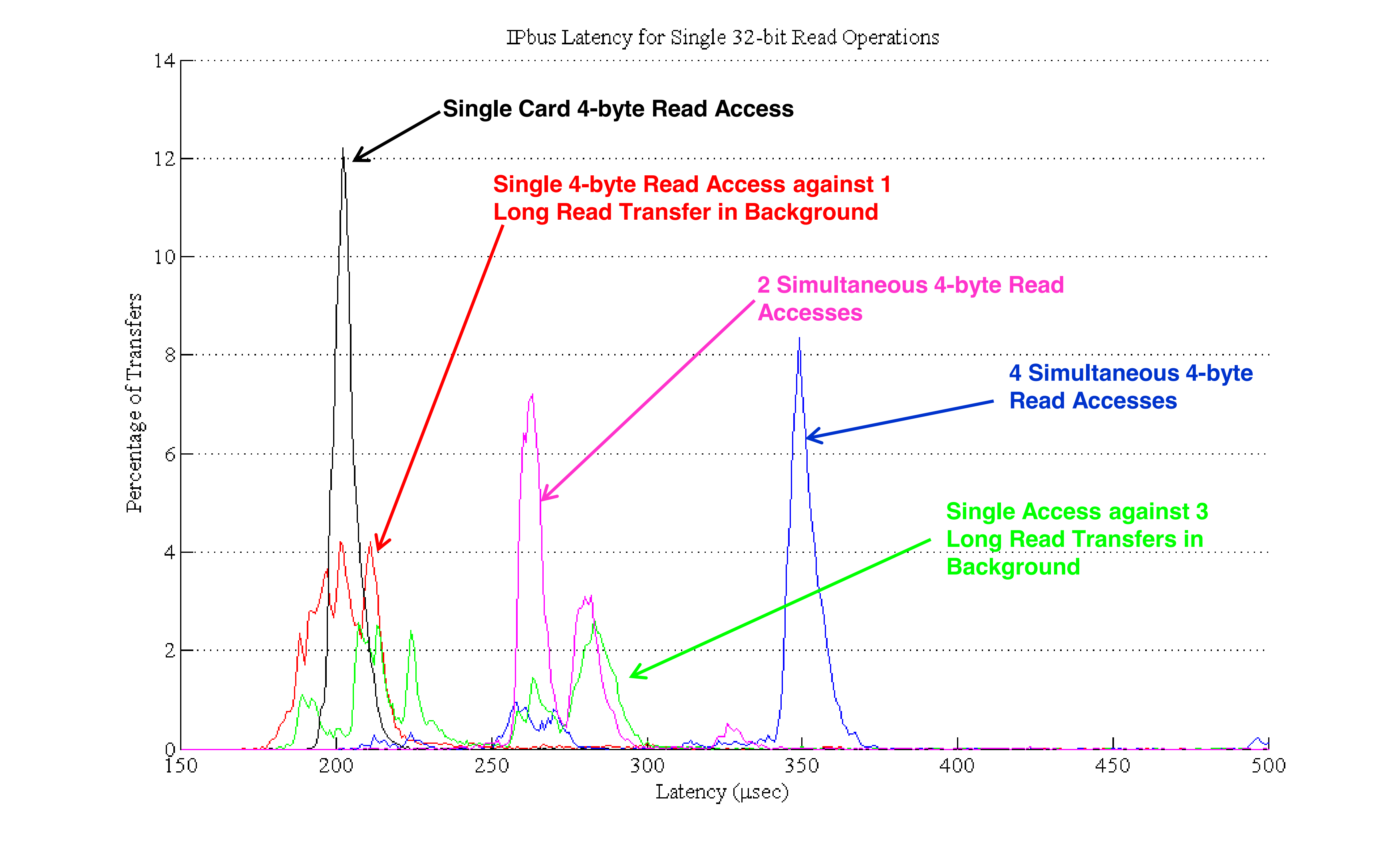
Some Common IPMI Command Interface Possibilities

- Option 1: Make a subset of an existing MMC (UW or other) command set common for CMS applications
  - Pros: Commands already defined & somewhat documented
  - Cons: Some aspects of commands specific to individual MMC implementations, possible conflicts between netFN or command codes between MMCs
- Option 2: Define a new netFN code (e.g., 34h/35h) as CMS netFN, and define common command set on “virgin soil”
  - Cons: new command set needs to be defined from ground up, possible near-duplication of commands in a given implementation
  - Pros: no conflicts with existing implementations, can expand over time

Summary

- UW MMC Version 2.0
  - Initial Testing to begin on UW CTP board
  - Then ported to BU AMC13—Summer 2012
- System Manager (PC-based IPMI component)
  - For laboratory benchtop development and installation use by users of UW MMC (short term & long term objectives)
  - Point of intersection between Run Control, Hardware & DCS
    - Possible opportunity for additional implementation or interface commonality
  - UW R&D Ongoing through summer/fall
- We expect to be in a position to tie into whatever common interfaces arise with a minimum of difficulty

IPbus Short Transfer Latency Measurements



Single Card 4-byte Read Access

Single 4-byte Read Access against 1 Long Read Transfer in Background

2 Simultaneous 4-byte Read Accesses

4 Simultaneous 4-byte Read Accesses

Single Access against 3 Long Read Transfers in Background

IPbus Short Transfer Latency Measurements

- **Test Hardware:**
  - Quad 2.1 GHz Xeon PC Running SL 5.3
  - GbE Capable NIC with 192.168.x.x network
  - NAT MCH with direct connection to NIC via Cat 5e cable
  - 4 UW Aux Cards (Virtex-5 FPGA)
- **Results:**
  - Measured “Best Case” Latency was ~200μs for single 4-byte read operation on otherwise “quiet” PC
  - Running 4 such reads in parallel, latency increased to ~350*μ*s on all 4 cards
  - Packet Level round trip time consistently ~45*μ*s (‘Hardware Latency’)

IPbus Short Transfer Latency Measurements, cont’d

- **Comparisons:**
  - CAEN PCI-VME Bridge: ~10*μ*s (mean measured value on installed RCT system)
  - Virtex-6 Microblaze: ~100ns
- **Observations**
  - For short transfers, IPbus Latency is greater than CAEN PCI-VME bridge currently in use, potentially by an order of magnitude or more
  - Minority of the latency is in the hardware/network

Appendix: UW MMC V 2.0 IPMI Command Summary

Get Device ID	Get FRU Inventory Area Info	Set Handle Override
Cold Reset	Read FRU Data	Set Current Requirement
Broadcast “Get Device ID”	Write FRU Data	Set Analog Scale Factor
Set Event Receiver	Set Backend Power	Get Analog Scale Factor
Get Event Receiver	Get Backend Power	Get Time Statistics
Platform Event (a.k.a. “Event Message”)	Set Payload Manager Settings	Set System Time
Get Device SDR Info	Get Payload Manager Settings	Get System Time
Get Device SDR	Get Fault Status	FPGA Configuration Read Status Register
Reserve Device SDR Repository	Set Boot Mode	FPGA Configuration Write Control Register
Get Sensor Hysteresis	Get Boot Mode	FPGA Configuration Write Data
Set Sensor Threshold	Set Backend Power Enable Mask	FPGA Configuration Read Data
Get Sensor Threshold	Get Backend Power Enable Mask	FPGA Configuration Nonvolatile Header Write
Set Sensor Event Enable	Set Sensor Alarm Mask	FPGA Configuration Nonvolatile Header Read
Get Sensor Event Enable	Get Sensor Alarm Mask	Get Nonvolatile Area Info
Get Sensor Reading		Raw Nonvolatile Write
Get PICMG Properties		Raw Nonvolatile Read
FRU Control		Check EEPROM Busy
Get FRU LED Properties		EEPROM Erase
Get LED Color Capabilities		
Set FRU LED State		
Get FRU LED State		
Get Device Locator Record ID		
FRU Control Capabilities		

NetFn Class Color Code

Application (06h/07h)

Sensor/Event (04h/05h)

Storage (0Ah/0Bh)

PICMG (2Ch/2Dh)

UWMMC (32h/33h)